UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/768,306	01/29/2004	John S. Worley	200208206-1	6318

22879 7590 09/07/2006

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
PEUGH, BRIAN R	

ART UNIT	PAPER NUMBER
2187	

DATE MAILED: 09/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/768,306	WORLEY, JOHN S.	
	Examiner	Art Unit	
	Brian R. Peugh	2187	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 June 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 June 2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Response to Amendment

This Office Action is in response to applicant's communication filed June 5, 2006 in response to PTO Office Action dated February 28, 2006. The applicant's remarks and amendment to the specification and/or claims were considered with the results that follow.

Claims 1-31 have been presented for examination in this application.

Please note the change in Examiner attributed to the current Application.

Drawings

The drawings are objected to under 37 CFR 1.83(a) because they fail to show the page allocation of step 316 as described in the specification. Any structural detail that is essential for a proper understanding of the disclosed invention should be shown in the drawing. MPEP § 608.02(d). Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after

the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 21-31 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims appear to disclose an operating system without a computer-readable medium needed to realize the operating system's functionality. Such a claimed operating system does not define any structural and functional interrelationships between the operating system and other claimed elements of a computer, which permit the operating system's functionality to be realized.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-2, 4, 13, 21-22 and 24 are rejected under 35 U.S.C. 102(b) as being anticipated by "Inside Windows NT Second Edition" by David A. Solomon (hereinafter referred to as Solomon).

Claim 1

A method for providing immediate virtual memory within a computer system **(Pg. 219, ¶s 1-2; It is noted that the reserved memory is the immediate virtual memory)**, the method comprising:

Allocating a new translation **(Pgs. 256-259 Page Table Entries (PTE), 273-274 Virtual Address Descriptors (VAD))** for a virtual memory page **(Pgs. 273, ¶4 and 274 ¶1; The allocation process consists of the allocation of the VAD and the PTE with both structures being part of the translation)**; and

Setting one or more bit flags within the translation to indicate that the translation specifies an immediate virtual page **(Pgs. 258-259 Fig. 5-11, Table 5-12 Valid bit)**.

Claim 2

The method of claim 1 wherein the new translation is a translation look-aside buffer entry allocated within a translation look-aside buffer **(inherent; Pg. 261, ¶1 under the Translation Look-Aside Buffer section; The reference states that the most recently used pages have entries in the TLB and a recently created entry is a most recently used entry and therefore would be in the TLB)**.

Claim 4

The method of claim 1 wherein the new translation is allocated within a memory-resident operating-system data structure **(Pgs. 256-259 Page Table Entries, 273-274**

Virtual Address Descriptors; The reference discloses allocating a Page Table Entry and a Virtual Address Descriptor, both of which are operating-system data structures).

Claim 13

A computer processor that provides architecture support for immediate virtual memory, the computer processor comprising:

Processor logic for executing computer instructions and fetching instructions and data from memory;

Processor logic for reading one or more control bits within a translation and determining whether or not a corresponding unit of memory is immediate **(inherent; Solomon discloses an operating system wherein the operating system allocates virtual memory and is able to differentiate between committed virtual memory and reserved (immediate) virtual memory through a valid bit in the translation of the virtual memory as discussed above with reference to claim 1. Solomon does not disclose expressly that there is a processor with processor logic for fetching and executing instructions and determining if a virtual memory allocation is an immediate virtual memory allocation, however a processor is inherently required for an operating system to operate. Since an operating system is merely software, and therefore consists of a series of instructions, the processor must have logic to fetch and execute these instructions. Furthermore, since the operating system is configured to operate on virtual memory, the processor logic must also fetch data from memory. Finally, since the operating system is**

configured to operate on immediate virtual memory and non-immediate virtual memory, which are distinguished by a bit in a translation, the processor must have logic that is able to read one or more control bits within the translation and determine whether or not the corresponding unit of memory is immediate).

Claim 21

An operating system that allocates an immediate virtual memory page within a computer system **(Pg. 219, ¶s 1-2; It is noted that the reserved memory is the immediate virtual memory)** by:

Allocating a new translation **(Pgs. 256-259 Page Table Entries (PTE), 273-274 Virtual Address Descriptors (VAD))** for the virtual memory page **(Pgs. 273, ¶4 and 274 ¶1; The allocation process consists of the allocation of the VAD and the PTE with both structures being part of the translation)**; and

Setting an immediate bit flag within the translation to indicate that the corresponding virtual memory page is immediate, with no allocated physical memory **(Pgs. 258-259 Fig. 5-11, Table 5-12 Valid bit)**.

Claim 22

The operating system of claim 21 wherein the new translation is a translation look-aside buffer entry allocated within a translation look-aside buffer **(inherent; Pg. 261, ¶1 under the Translation Look-Aside Buffer section; The reference states that the most recently used pages have entries in the TLB and a recently created entry is a most recently used entry and therefore would be in the TLB)**.

Claim 24

The operating system of claim 21 wherein the new translation is allocated within a memory-resident operating-system data structure **(Pgs. 256-259 Page Table Entries, 273-274 Virtual Address Descriptors; The reference discloses allocating a Page Table Entry and a Virtual Address Descriptor, both of which are operating-system data structures).**

Claims 5, 8-9, 12, 14, 17, 20, 25, 28-29, and 31 are rejected under 35 U.S.C. 102(b) as being anticipated by Solomon in view of <http://squid-cache.org/mail-archive/squi-users/199708/0124.html> (hereinafter referred to as Wemm) with the latter being provided in accordance with MPEP 2131.01(II).

Claim 5

The method of claim 1 wherein, when immediate virtual memory is accessed by a READ access instruction, a specified value is returned **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that the demand-zero response is a page fault response that occurs when a newly allocated page is accessed for the first time and results in the allocating and returning of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation).**

Claim 8

The method of claim 5 wherein the specified value is specified by one of:

A software specification within an operating system; or

A hardware logic circuit **(Solomon, Pg. 266 – Demand Zero; Solomon discloses wherein the specified value is chosen from a list or generated both of which are performed by a software specification within an operating system)**.

Claim 9

The method of claim 5 wherein the specified value is 0 **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that demand-zero response is a page fault response that occurs when a newly allocated page is accessed for the first time and results in the allocating and returning of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation)**.

Claim 12

The method of claim 1 wherein, when immediate virtual memory is accessed by a WRITE access instruction, an exception is generated to allow an operating system to allocate and initialize a physical memory page corresponding to the virtual memory page **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault (generated exception) wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that demand-zero response is a page fault response that occurs when a newly allocated page is accessed for the first time and results in the allocating and**

returning of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation).

Claim 14

The computer processor of claim 13 further including,

Processor logic that, upon READ access to memory determined to be immediate, returns a specified value; and

Processor logic that, upon WRITE access to immediate memory, generates an immediate memory exception to allow an operating system to allocate and initialize physical memory corresponding to the immediate memory **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that demand-zero response is a page fault response that occurs when a newly allocated page is accessed for the first time and results in the returning/allocating of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation. With regards to the processor logic, see the discussion for claim 13 above).**

Claim 17

The computer processor of claim 14 wherein the specified value is 0 **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that demand-zero response is a page fault**

response that occurs when a newly allocated page is accessed for the first time and results in the returning of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation).

Claim 20

The computer processor of claim 14 wherein the specified value is specified by one of:

A software specification within an operating system; or

A hardware logic circuit **(Solomon, Pg. 266 – Demand Zero; Solomon discloses wherein the specified value is chosen from a list or generated both of which are performed by a software specification within an operating system).**

Claim 25

The operating system of claim 21 wherein, when immediate virtual memory is accessed by a READ access instruction, a specified value is returned **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that the demand-zero response is a page fault response that occurs when a newly allocated page is accessed for the first time and results in the allocating and returning of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation).**

Claim 28

The operating system of claim 25 wherein the specified value is 0 **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that demand-zero response is a page fault response that occurs when a newly allocated page is accessed for the first time and results in the allocating and returning of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation).**

Claim 31

The operating system of claim 21 wherein, when immediate virtual memory is accessed by a WRITE access instruction, an immediate-virtual-memory-page exception is generated to allow the operating system to allocate and initialize a physical memory page corresponding to the virtual memory page **(Solomon, Pg. 266 – Demand Zero; As disclosed by Solomon, the demand-zero response is a response to a page fault (generated exception) wherein the desired page must be satisfied with a page of zeros. Wemm discloses in ¶4 that demand-zero response is a page fault response that occurs when a newly allocated page is accessed for the first time and results in the allocating and returning of a zeroed page. The newly allocated page has been reserved but does not have a physical page attached to it upon allocation).**

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 2 and 22 are rejected under 35 U.S.C. 102(b) as anticipated by or, in the alternative, under 35 U.S.C. 103(a) as obvious over Solomon in view of applicant's admitted prior art (hereinafter referred to as AAPA).

Claim 2 – 103(a)

(Note: See 102(b) rejection in ¶8 above)

Solomon discloses the method of claim 1 as above.

Solomon does not disclose expressly wherein the new translation is a translation look-aside buffer (TLB) entry allocated within a translation look-aside buffer.

AAPA discloses that when a virtual memory page is allocated, corresponding TLB entries are placed into a TLB **(Pg. 4, Lines 5-7)**.

Solomon and AAPA are analogous art because they are from the same field of endeavor of allocating virtual memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of Solomon and AAPA before them, to place translations of a virtual memory into a TLB.

The motivation for doing so would have been to use the speed of TLB registers to prevent decrease in system performance due to translation from virtual to physical

addresses **(Solomon, Pg. 261, ¶1 under the Translation Look-Aside Buffer section)**.

Therefore, it would have been obvious to combine AAPA with Solomon for the benefit of improved system performance to obtain the invention as specified in claim 2.

Claim 22 – 103(a)

(Note: See 102(b) rejection in ¶8 above)

Solomon discloses the operating system of claim 21 as above.

Solomon does not disclose expressly wherein the new translation is a translation look-aside buffer (TLB) entry allocated within a translation look-aside buffer.

AAPA discloses that when a virtual memory page is allocated, corresponding TLB entries are placed into a TLB **(Pg. 4, Lines 5-7)**.

Solomon and AAPA are analogous art because they are from the same field of endeavor of allocating virtual memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of Solomon and AAPA before them, to place translations of a virtual memory into a TLB.

The motivation for doing so would have been to use the speed of TLB registers to prevent decrease in system performance due to translation from virtual to physical addresses **(Solomon, Pg. 261, ¶1 under the Translation Look-Aside Buffer section)**.

Therefore, it would have been obvious to combine AAPA with Solomon for the benefit of improved system performance to obtain the invention as specified in claim 22.

Claims 3 and 23 are rejected under 35 U.S.C. 103(a) as being obvious over Solomon in view of AAPA.

Claim 3

Solomon discloses the method of claim 2 as above.

Solomon does not disclose expressly wherein the new translation look-aside buffer entry is a translation look-aside buffer entry allocated within a virtual hash page table.

AAPA discloses wherein a TLB entry is stored in a virtual hash page table **(Pg. 2, Lines 20-21)**.

Solomon and AAPA are analogous art because they are from the same field of endeavor of allocating virtual memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of Solomon and AAPA before them, to place TLB entries into a virtual hash page table (VHPT).

The motivation for doing so would have been to ensure TLB entries are backed up so they are not lost **(AAPA, Page 2, Lines 20-21)**.

Therefore, it would have been obvious to combine AAPA with Solomon for the benefit of improved system performance to obtain the invention as specified in claim 3.

Claim 23

Solomon discloses the operating system of claim 22 as above.

Solomon does not disclose expressly wherein the new translation look-aside buffer entry is a translation look-aside buffer entry allocated within a virtual hash page table.

AAPA discloses wherein a TLB entry is stored in a virtual hash page table **(Pg. 2, Lines 20-21)**.

Solomon and AAPA are analogous art because they are from the same field of endeavor of allocating virtual memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of Solomon and AAPA before them, to place TLB entries into a virtual hash page table (VHPT).

The motivation for doing so would have been to ensure TLB entries are backed up so they are not lost **(AAPA, Page 2, Lines 20-21)**.

Therefore, it would have been obvious to combine AAPA with Solomon for the benefit of improved system performance to obtain the invention as specified in claim 23.

Claims 6-7, 15-16 and 26-27 are rejected under 35 U.S.C. 103(a) as being obvious over Solomon in view of Wemm with the latter being provided in accordance with MPEP 2131.01(II) as applied to claim 5 above and further in view of "Structured Computer Organization Second Edition" by Andrew S. Tanenbaum (hereinafter referred to as Tanenbaum).

Claim 6

The combination of Solomon and Wemm disclose the method of claim 5 as above. Solomon further discloses wherein a specified value is either obtained from a list or generated by software **(Solomon, Pg. 266 – Demand Zero)**.

They do not disclose expressly wherein the specified value is generated by a processor logic circuit.

Tanenbaum discloses wherein hardware and software are logically equivalent and therefore any operation performed by software can also be built directly into the hardware **(Pg. 11, Line 11)**. Therefore, having the specified value generated by software is logically equivalent to having the value generated by a processor logic circuit.

The combination of Solomon and Wemm and Tanenbaum are analogous art because they are from the same field of endeavor of computer hardware.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Tanenbaum before them, to use a processor logic circuit to generate a specified return value in response to a read command.

The motivation for doing so would have been to optimize such factors as cost, speed and reliability **(Tanenbaum, Pg. 11, Lines 14-15)**.

Therefore, it would have been obvious to combine the combination of Solomon and Wemm with Tanenbaum for the benefit of obtaining the invention as specified in claim 6.

Claim 7

The combination of Solomon and Wemm disclose the method of claim 5 as above. Solomon further discloses wherein a specified value is either obtained from a list or generated by software **(Solomon, Pg. 266 – Demand Zero)**.

They do not disclose expressly wherein the specified value is obtained from a default-valued processor register.

Tanenbaum discloses wherein hardware and software are logically equivalent and therefore any operation performed by software can also be built directly into the hardware **(Pg. 11, Line 11)**. Therefore, having the specified value obtained from a list is logically equivalent to having the value obtained from a processor register.

The combination of Solomon and Wemm and Tanenbaum are analogous art because they are from the same field of endeavor of computer hardware.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Tanenbaum before them, to use a processor logic circuit to generate a specified return value in response to a read command.

The motivation for doing so would have been to optimize such factors as cost, speed and reliability **(Tanenbaum, Pg. 11, Lines 14-15)**.

Therefore, it would have been obvious to combine the combination of Solomon and Wemm with Tanenbaum for the benefit of obtaining the invention as specified in claim 7.

Claim 15

The combination of Solomon and Wemm disclose the computer processor of claim 14 as above. Solomon further discloses wherein a specified value is either obtained from a list or generated by software **(Solomon, Pg. 266 – Demand Zero)**.

They do not disclose expressly wherein the specified value is generated by a processor logic circuit.

Tanenbaum discloses wherein hardware and software are logically equivalent and therefore any operation performed by software can also be built directly into the hardware **(Pg. 11, Line 11)**. Therefore, having the specified value generated by software is logically equivalent to having the value generated by a processor logic circuit.

The combination of Solomon and Wemm and Tanenbaum are analogous art because they are from the same field of endeavor of computer hardware.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Tanenbaum before them, to use a processor logic circuit to generate a specified return value in response to a read command.

The motivation for doing so would have been to optimize such factors as cost, speed and reliability **(Tanenbaum, Pg. 11, Lines 14-15)**.

Therefore, it would have been obvious to combine the combination of Solomon and Wemm with Tanenbaum for the benefit of obtaining the invention as specified in claim 15.

Claim 16

The combination of Solomon and Wemm disclose the computer processor of claim 14 as above. Solomon further discloses wherein a specified value is either obtained from a list or generated by software **(Solomon, Pg. 266 – Demand Zero)**.

They do not disclose expressly wherein the specified value is obtained from a default-valued processor register.

Tanenbaum discloses wherein hardware and software are logically equivalent and therefore any operation performed by software can also be built directly into the hardware **(Pg. 11, Line 11)**. Therefore, having the specified value obtained from a list is logically equivalent to having the value obtained from a processor register.

The combination of Solomon and Wemm and Tanenbaum are analogous art because they are from the same field of endeavor of computer hardware.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Tanenbaum before them, to use a processor logic circuit to generate a specified return value in response to a read command.

The motivation for doing so would have been to optimize such factors as cost, speed and reliability **(Tanenbaum, Pg. 11, Lines 14-15)**.

Therefore, it would have been obvious to combine the combination of Solomon and Wemm with Tanenbaum for the benefit of obtaining the invention as specified in claim 16.

Claim 26

The combination of Solomon and Wemm disclose the operating system of claim 25 as above. Solomon further discloses wherein a specified value is either obtained from a list or generated by software **(Solomon, Pg. 266 – Demand Zero)**.

They do not disclose expressly wherein the specified value is generated by a processor logic circuit.

Tanenbaum discloses wherein hardware and software are logically equivalent and therefore any operation performed by software can also be built directly into the hardware **(Pg. 11, Line 11)**. Therefore, having the specified value generated by software is logically equivalent to having the value generated by a processor logic circuit.

The combination of Solomon and Wemm and Tanenbaum are analogous art because they are from the same field of endeavor of computer hardware.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Tanenbaum before them, to use a processor logic circuit to generate a specified return value in response to a read command.

The motivation for doing so would have been to optimize such factors as cost, speed and reliability **(Tanenbaum, Pg. 11, Lines 14-15)**.

Therefore, it would have been obvious to combine the combination of Solomon and Wemm with Tanenbaum for the benefit of obtaining the invention as specified in claim 26.

Claim 27

The combination of Solomon and Wemm disclose the operating system of claim 25 as above. Solomon further discloses wherein a specified value is either obtained from a list or generated by software **(Solomon, Pg. 266 – Demand Zero)**.

They do not disclose expressly wherein the specified value is obtained from a default-valued processor register.

Tanenbaum discloses wherein hardware and software are logically equivalent and therefore any operation performed by software can also be built directly into the hardware **(Pg. 11, Line 11)**. Therefore, having the specified value obtained from a list is logically equivalent to having the value obtained from a processor register.

The combination of Solomon and Wemm and Tanenbaum are analogous art because they are from the same field of endeavor of computer hardware.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Tanenbaum before them, to use a processor logic circuit to generate a specified return value in response to a read command.

The motivation for doing so would have been to optimize such factors as cost, speed and reliability **(Tanenbaum, Pg. 11, Lines 14-15)**.

Therefore, it would have been obvious to combine the combination of Solomon and Wemm with Tanenbaum for the benefit of obtaining the invention as specified in claim 27.

Claims 10, 18 and 30 are rejected under 35 U.S.C. 103(a) as being obvious over Solomon in view of Wemm with the latter being provided in accordance with MPEP 2131.01(II) as applied to claims 5, 14 and 25 above, and further in view of LeClerg (6,857,041).

Claim 10

The combination of Solomon and Wemm disclose the method of claim 5 as above.

They do not disclose expressly, wherein the specified value is any fixed, non-zero bit pattern of any size.

LeClerg discloses a method for initializing memory wherein the memory may be initialized to one. In an alternate embodiment, the memory may be initialized to any other suitable value **(Col. 3, Lines 25-34; It is noted that LeClerg discloses the initialization of physical memory and not virtual memory, however the examiner is interpreting initialization of virtual pages as disclosed on line 17, page 7 of the instant specification, as initialization of the physical memory associated with the virtual memory. Since LeClerg specifically discloses embodiments where the memory is initialized to either one or zero, the examiner is interpreting "any other suitable value" as any value other than one or zero)**.

The combination of Solomon and Wemm and LeClerg are analogous art because they are from the same field of endeavor of initializing memory in varying manners.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and LeClerg before them, to initialize memory to any fixed, non-zero bit pattern of any size.

The motivation for doing so would have been to erase secure information from the memory so that it may not be accessed by others **(LeClerg, Col. 1, Lines 22-33)**.

Therefore, it would have been obvious to combine LeClerg with the combination of Solomon and Wemm to obtain the invention as specified in claim 10.

Claim 18

The combination of Solomon and Wemm disclose the computer processor of claim 14 as above.

They do not disclose expressly, wherein the specified value is any fixed, non-zero bit pattern of any size.

LeClerg discloses a method for initializing memory wherein the memory may be initialized to one. In an alternate embodiment, the memory may be initialized to any other suitable value **(Col. 3, Lines 25-34; It is noted that LeClerg discloses the initialization of physical memory and not virtual memory, however the examiner is interpreting initialization of virtual pages as disclosed on line 17, page 7 of the instant specification, as initialization of the physical memory associated with the virtual memory. Since LeClerg specifically discloses embodiments where the memory is initialized to either one or zero, the examiner is interpreting "any other suitable value" as any value other than one or zero)**.

The combination of Solomon and Wemm and LeClerg are analogous art because they are from the same field of endeavor of initializing memory in varying manners.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and LeClerg before them, to initialize memory to any fixed, non-zero bit pattern of any size.

The motivation for doing so would have been to erase secure information from the memory so that it may not be accessed by others **(LeClerg, Col. 1, Lines 22-33)**.

Therefore, it would have been obvious to combine LeClerg with the combination of Solomon and Wemm to obtain the invention as specified in claim 18.

Claim 30

The combination of Solomon and Wemm disclose the operating system of claim 25 as above.

They do not disclose expressly wherein the specified value is a random number obtained by processor logic algorithmically, from electronic noise, or from another physical source.

LeClerg discloses a method for initializing memory wherein the memory may be initialized to any other suitable value. **(Col. 3, Lines 25-34; It is noted that LeClerg discloses the initialization of physical memory and not virtual memory, however the examiner is interpreting initialization of virtual pages as disclosed on line 17, page 7 of the instant specification, as initialization of the physical memory associated with the virtual memory. Since LeClerg specifically discloses**

embodiments where the memory is initialized to either one or zero, the examiner is interpreting "any other suitable value" as any value other than one or zero).

The combination of Solomon and Wemm and LeClerg are analogous art because they are from the same field of endeavor of initializing memory in varying manners.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and LeClerg before them, to initialize memory to a random number.

The motivation for doing so would have been to erase secure information from the memory so that it may not be accessed by others **(LeClerg, Col. 1, Lines 22-33).**

Therefore, it would have been obvious to combine LeClerg with the combination of Solomon and Wemm to obtain the invention as specified in claim 30.

Claims 11 and 19 are rejected under 35 U.S.C. 103(a) as being obvious over Solomon in view of Wemm with the latter being provided in accordance with MPEP 2131.01(II) as applied to claims 5 and 14 above, and further in view of Liew (6,665,249).

Claim 11

The combination of Solomon and Wemm disclose the method of claim 5 as above.

They do not disclose expressly wherein the specified value is a random number obtained by processor logic algorithmically, from electronic noise, or from another physical source.

Liew discloses a method for initializing memory wherein the memory is initialized to a random value **(Title; Abstract; It is noted that Liew discloses the initialization**

of physical memory and not virtual memory, however the examiner is interpreting initialization of virtual pages as disclosed on line 17, page 7 of the instant specification, as initialization of the physical memory associated with the virtual memory.).

The combination of Solomon and Wemm and Liew are analogous art because they are from the same field of endeavor of methods for initializing memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Liew before them, to initialize memory to a random number.

The motivation for doing so would have been to improve the selection of the optimum write power and thus provide increased reliability and performance **(Liew, Col. 1, Lines 49-56)**.

Therefore, it would have been obvious to combine Liew with the combination of Solomon and Wemm to obtain the invention as specified in claim 11.

Claim 19

The combination of Solomon and Wemm disclose the computer processor of claim 14 as above.

They do not disclose expressly wherein the specified value is a random number obtained by processor logic algorithmically, from electronic noise, or from another physical source.

Liew discloses a method for initializing memory wherein the memory is initialized to a random value **(Title; Abstract; It is noted that Liew discloses the initialization**

of physical memory and not virtual memory, however the examiner is interpreting initialization of virtual pages as disclosed on line 17, page 7 of the instant specification, as initialization of the physical memory associated with the virtual memory.).

The combination of Solomon and Wemm and Liew are analogous art because they are from the same field of endeavor of methods for initializing memory.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Liew before them, to initialize memory to a random number.

The motivation for doing so would have been to improve the selection of the optimum write power and thus provide increased reliability and performance **(Liew, Col. 1, Lines 49-56)**.

Therefore, it would have been obvious to combine Liew with the combination of Solomon and Wemm to obtain the invention as specified in claim 19.

Claim 29 is rejected under 35 U.S.C. 103(a) as being obvious over Solomon in view of Wemm with the latter being provided in accordance with MPEP 2131.01(II) as applied to claim 25 above, and further in view of Sakakura et al. (5,625,795; Hereinafter referred to as Sakakura) and <http://www.cs.jcu.edu.au/Subjects/cp1200/1996/org/node9.html> (hereinafter referred to as Sloane).

Claim 29

The combination of Solomon and Wemm disclose the operating system of claim 25 as above.

They do not disclose expressly, wherein the specified value is –1 in two's complement arithmetic.

Sakakura discloses initializing a memory wherein the memory is initialized to –1 **(Col. 7, Lines 16-20; It is noted that Sakakura discloses the initialization of physical memory and not virtual memory, however the examiner is interpreting initialization of virtual pages as disclosed on line 17, page 7 of the instant specification, as initialization of the physical memory associated with the virtual memory.)**.

The combination of Solomon and Wemm and LeClerg are analogous art because they are from the similar problem solving area of indicating certain memory area as unusable.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and Liew before them, to initialize memory to -1.

The motivation for doing so would have been to improve reliability of exclusive control over the memory **(Sakakura, Col. 2, Lines 40-45)**.

Therefore, it would have been obvious to combine Sakakura with the combination of Solomon and Wemm to obtain the invention as specified in claim 29.

The combination of Solomon and Wemm and LeClerg does not disclose expressly where –1 is represented in two's complement.

Sloane discloses representing numbers in two's complement.

At the time of the invention it would have been obvious to a person of ordinary skill in the art, having the teachings of the combination of Solomon and Wemm and LeClerg and Sloane before them, to represent –1 as a two's complement number.

The motivation for doing so would have been due to the fact that two's complement achieves subtraction in the same manner as addition.

Therefore, it would have been obvious to combine Sloane with the combination of Solomon and Wemm and LeClerg to obtain the invention as disclosed in claim 29.

Response to Arguments

Applicant's arguments filed June 5, 2006 have been fully considered but they are not persuasive.

Regarding Applicant's argument of page 16, paragraph 2, that Figure 3 does not omit any processing steps as recited in the Specification, the Examiner disagrees. Figure 3, step 318 does not recite the allocation and accessing of a page (see Applicant's response, page 8, lines 14-18).

Regarding Applicant's Argument directed towards the 35 U.S.C. 101 rejection of page 17, the Examiner disagrees. An Operating System has been claimed, however a

statutory medium upon which the OS resides, and from which it must be executed from, has not been recited.

Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Applicant's arguments of pages 4-16 separately describe the claimed invention and prior art references, however there appears to be no arguments separately addressing the individual claims (and associated limitations) with the respective prior art rejections as cited by the Examiner.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Brian R. Peugh whose telephone number is (571) 272-4199. The examiner can normally be reached on Monday-Thursday from 7:00am to 4:30pm. The examiner can also be reached on alternate Friday's from 7:00am to 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks, can be reached on (571) 272-4201. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Brian R. Peugh
Primary Examiner
8/31/06